Anomaly Detection in GPS Data Based on Visual Analytics

Zicheng Liao * Yizhou Yu † Baoquan Chen ‡

* † University of Illinois at Urbana-Champaign
‡ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

ABSTRACT

Modern machine learning techniques provide robust approaches for data-driven modeling and critical information extraction, while human experts hold the advantage of possessing high-level intelligence and domain-specific expertise. We combine the power of the two for anomaly detection in GPS data by integrating them through a visualization and human-computer interaction interface.

In this paper we introduce GPSvas (GPS Visual Analytics System), a system that detects anomalies in GPS data using the approach of visual analytics: a conditional random field (CRF) model is used as the machine learning component for anomaly detection in streaming GPS traces. A visualization component and an interactive user interface are built to visualize the data stream, display significant analysis results (i.e., anomalies or uncertain predications) and hidden information extracted by the anomaly detection model, which enable human experts to observe the real-time data behavior and gain insights into the data flow. Human experts further provide guidance to the machine learning model through the interaction tools; the learning model is then incrementally improved through an active learning procedure.

Index Terms: H.1.2 [Models and Principles]: User/Machine Systems—Human information processing; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphics user interfaces; I.5.2 [Pattern Recognition]: Design Methodology—Pattern analysis, Feature evaluation and selection;

1 INTRODUCTION

With the prevalence of the Global Positioning System (GPS), an increasing number of electronic devices and vehicles have been equipped with a GPS module for a variety of applications including navigation and location-based search. In addition to such conventional use, the GPS module can also be treated as a sensor that can regularly report the position and other status of the hosting vehicle or object. Such GPS traces provide very useful information regarding the temporal trajectory and the moving pattern of the host as well as indirect information regarding the surroundings of the host. In this paper, we focus on data analytics and anomaly detection of GPS traces of urban taxis.

There are three major objectives of such data analysis: 1) use taxi GPS traces to assist urban traffic monitoring because the speed of a taxi indirectly indicates the traffic condition on the street where the taxi is; 2) improve the safety of pedestrians and taxi passengers by monitoring and detecting reckless behaviors of taxi drivers; 3) discover potential emergencies or abnormal situations associated with taxi drivers or passengers. There exist a few challenges to achieve these objectives. First, we need to deal with a large number of simultaneous real-time data streams because there are typically a large number of taxis in an urban area; second, we need to efficiently analyze the temporal patterns of individual GPS traces as well as spatial distributions of these traces to report any abnormal traffic conditions or driving behaviors in real-time.

Manually analyzing hundreds of GPS traces is obviously unrealistic. On the other hand, a completely automatic approach would not be feasible either since abnormal situations need to be confirmed by human experts. Therefore, a visual analytics approach is taken to develop a semi-automatic system. There should exist both data analysis and visualization components in the system to support collaboration between machines and human analysts. The data analysis component is based on machine learning models. Fast automatic analysis is first performed by the data analysis component, which is also capable of providing the uncertainty of the analysis results. GPS traces along with analysis results are presented through the visualization engine. Human analysts can make use of the visualization in multiple different ways. Most of the time, the automatic analysis results are correct with high confidence. Therefore, human analysts can directly take the results provided by the machine. When a result is presented with high uncertainty, a human analyst can interact with our system to look at the details of the spatial and temporal patterns to correct the automatic analysis result. More importantly, any user-provided analysis results can also be used as training data to improve the performance of the automatic data analysis component so that it can achieve a higher level of accuracy on future incoming data.

Our system harnesses the computing power of machine learning models and the high-level intelligence of human experts through a visualization engine and a human-computer interaction interface. Through the visualization and the interaction tools, human experts can choose to browse the most relevant information and provide guidance to the anomaly detection component. We use a state-of-the-art discriminative machine learning model, conditional random fields (CRFs), for anomaly detection. CRFs require supervised training. To minimize the amount of manual labeling for training the CRF model, the performance of our CRF model is incrementally improved through an active learning procedure. Active learning is a machine learning paradigm that the learner (machine model) selectively choose data items with a high prediction uncertainty as training set. This is because such data items are the most critical ones that can directly remove ambiguities in the machine model and effectively improve its performance.

The rest of this paper is organized as follows. In Section 2, we give an overview of our visual analytics system. In Section 3, we first give an overview of conditional random fields and then discuss in detail how to perform feature extraction in our CRF model for GPS anomaly detection. In Section 4, we first review the active learning approach, and then present the criteria we use to select candidate training data. In Section 5, we present our visualization component and the human-machine interface of our system. After that, we demonstrate experimental results of our system on a set of GPS traces in Section 6. Related work and conclusions are given in Sections 7 and 8.

*liao17@illinois.edu
†yyz@illinois.edu

IEEE Symposium on Visual Analytics Science and Technology
October 24 - 29, Salt Lake City, Utah, USA
978-1-4244-9487-3/10/$26.00 ©2010 IEEE

2 Overview

GPS traces from hundreds of taxis within an urban area serve as the input to our system. Such data can be streamed to our system in real-time. Data is automatically collected from every taxi once every few seconds. These collected data items form the trajectory of a taxi over time. A data item consists of 6 attributes: $(ID, latitude, longitude, loaded, speed, time)$. *ID* is the identification number of the taxi from which the data is collected. *latitude* and *longitude* define the global location of the taxi. *loaded* is a boolean value indicating whether the taxi is loaded with passengers or not. *speed* is simply the speed of the taxi at the time of collection. *time* is the time stamp of the GPS data item.

Our system consists of four major components: a machine model for anomaly detection, an active learning module, a visualization component, and a human-machine interaction interface. Figure 1 gives an overview of the system architecture.

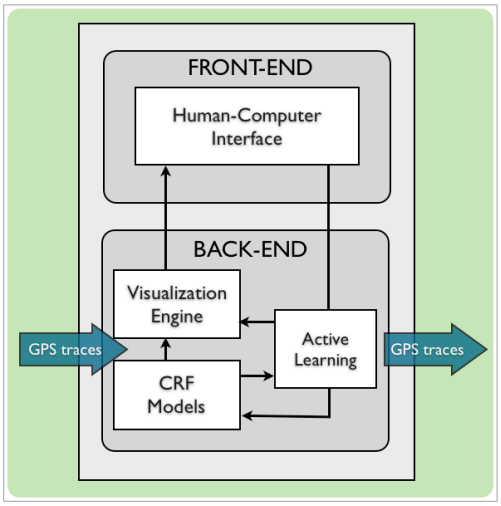


Figure 1: Overview of the system architecture

The human-machine interaction interface forms the front-end of the system, while the back-end consists of the visualization engine, the active learning module and an anomaly detection component based on the conditional random field (CRF) model. Since CRFs perform supervised classification, an initial CRF model with a reasonable classification accuracy needs to be trained in advance. The interaction interface supports three modes: *basic mode*, *monitoring mode* and *tagging mode*. A user can switch among these three modes at any time.

The basic mode only visualizes the raw GPS traces, and users can only perform basic exploration.

In the monitoring mode, the anomaly detection component is activated, and anomaly tags are shown dynamically on the screen. As shown in Figure 1, data passes through the visualization engine and the CRF model. At every time step, the CRF model predicts the status (normal or abnormal) of every taxi by analyzing the new incoming data together with previous data falling within a causal time window. The visualization engine takes the incoming GPS traces and the predicted labels from the CRF model to update the visualization on the screen. In addition, upon request from the user, the visualization engine can also show internal feature values used by the CRF model. Thus, human experts can not only verify the final analysis results from the CRF model but also gain additional insights by checking the evidences the CRF model uses to reach its conclusions.

In the tagging mode, the active learning module is activated. It uses the CRF model to mark data items whose labels are highly uncertain. High uncertainty indicates the current version of the CRF model has become inadequate to label these data items automatically. Human experts are requested to manually label a representative subset of these marked items. Such labeled data can then be used to train an improved version of the CRF model.

The visual analytics approach taken in our system gives a novel anomaly detection framework for GPS data with minimal user intervention by effectively integrating automatic state-of-the-art machine learning techniques with human experts' insights. Figure 2 shows a snapshot of the system. Note that even though the data flow and labels are visualized in real-time, the system allows users to roll-back to previous time stamps to re-check a label or to examine event logs offline in case that simultaneously occurring abnormal events cannot be handled in time.

3 Anomaly Detection Based on CRFs

Conditional random fields is a machine learning model for representing the conditional probability distribution of hidden states $\mathbf{Y}$ given observations $\mathbf{X}$. Intuitively, a conditional random field model builds the conditional probability distribution $P(\mathbf{Y}|\mathbf{X})$ for determining the most probable labeling of observation data. It was first introduced by Lafferty *et al.* [13] for text sequence segmentation and labeling, and has been successfully applied to many problems in text processing, such as part-of-speech (POS) tagging [13] and name entity recognition (NER) [18], as well as problems in other fields, such as bioinformatics [24] and computer vision [12].

In the remainder of the paper we limit our discussion of CRFs to linear-chain CRFs, which is the model we use for our GPS data anomaly detection.

3.1 Linear-Chain Conditional Random Fields

The state variables $\mathbf{Y}$ in a linear-chain conditional random field [13, 27] are restricted to form a chain. This assumption greatly simplifies the model complexity and yet is a very natural assumption in applications where the input $\mathbf{X}$ has a sequential form, such as text sequences for natural language processing problems or gene sequences for bioinformatics problems. Linear-chain condition random fields are well suited for modeling and classifying GPS data too because GPS data streams are essentially temporal signals and have a sequential nature.

The graphical representation of a linear-chain CRF is shown in Figure 3. Y_t is the hidden state variable for the node at position t in the sequence, $\mathbf{X}_t$ is the observed data at position t. Each state variable Y_t is only connected to the immediately preceding and following states (restriction of feature definition). The probability distribution over the random variables $(\mathbf{X}, \mathbf{Y})$ is modeled after a *Markov random field* (undirected graphical model), as is shown in Figure 3. By the fundamental theorem of Markov random fields [8], which states that the joint probability of a Markov random field can be factorized into a product of potential functions over local *cliques* (features) in the graph, the joint probability of the hidden state sequence $\mathbf{y}$ conditioned on $\mathbf{x}$ can be written in the following form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{\sum_{t=1}^{N}\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\}$$

where $Z(\mathbf{x})$ is the normalization item:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\{\sum_{t=1}^{N}\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\}$$

We use lowercase notations $\mathbf{x}, \mathbf{y}$ for assignments to the variables. N is the length of the input sequence and K is the size of the feature function set. Each f_k is a feature function (potential function) that is defined over a local clique in the graph, and takes y_t, y_{t-1} and items in the input observation sequence as arguments. A weight vector

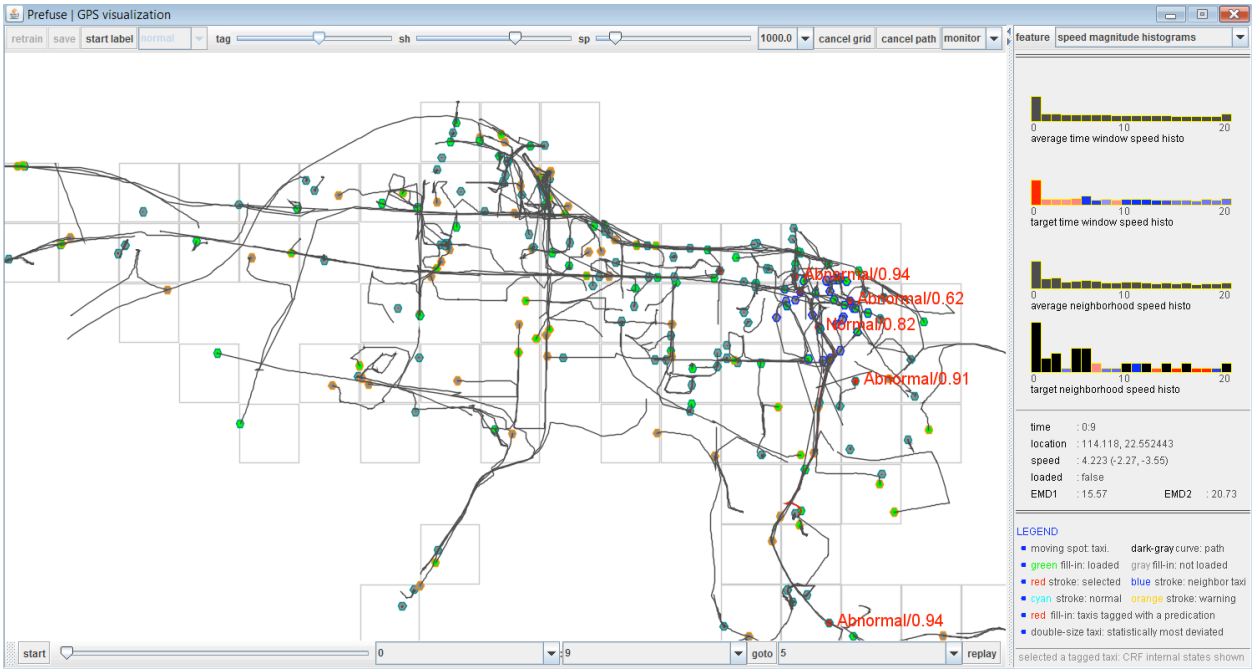


Figure 2: A snapshot of the visual interface. Middle left: main window for visualization and information display. Upper left: interface for visualization configurations and user operations. Bottom Left: interface that shows time and supports play-back. Right panel. Display dynamic histograms of the traffic or the selected items, a legend and system status.

$\lambda = \{\lambda_1, ..., \lambda_k\}$ is associated with the feature function set. The values in the weight vector determine how the feature functions contribute to the conditional probability computed by the model. A feature function $f_k(y_t, t_{t-1}, \mathbf{x}_t)$ is an indicator function that describes a local pattern, for example $f_k(y_t = abnormal, y_{t-1} = normal)$ returns 1 if the two state conditions are satisfied. The feature functions, $\{f_1, ..., f_k\}$, define a set of clique templates for the CRF model: given an input sequence $\mathbf{x}$, the graphical model for the sequence can be constructed by moving the feature templates over the sequence. Therefore, a CRF model is fully specified by the feature function set and the weight vector $(\mathbf{f}, \lambda)$.

Since a conditional random field is a supervised machine learning model, it needs to go through a supervised training stage before being used for new testing data.

• **training**: The training process computes the model parameters (the weight vector) according to labeled training data pairs $\{\mathbf{y}, \mathbf{x}\}^m$ such that the log-likelihood

$$\sum_{i=1}^{m} \log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = \sum_{i=1}^{m} \sum_{t=1}^{N} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_t^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{m} \log Z(\mathbf{x}^{(i)})$$

is maximized. The above objective function is convex so gradient-based optimization can guarantee to find the globally optimal solution. The state-of-the-art gradient ascent method for such an optimization problem is the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [16].

• **inference**: When a trained CRF model is applied to a novel input sequence, it tries to find the most likely hidden state assignment $\mathbf{y}$, i.e., the label sequence

$$\mathbf{y} = arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

for the unlabeled input sequence $\mathbf{x}$. For linear-chain CRFs, this can be efficiently performed by dynamic programming (the *Viterbi* algorithm [19]) over the sequential hidden variables $\mathbf{y}$.

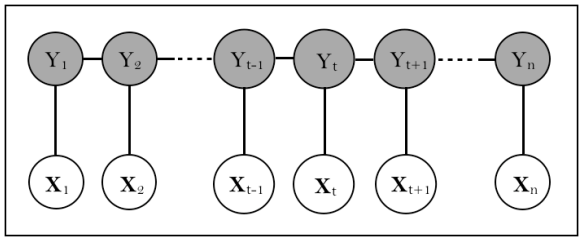


Figure 3: A graphical representation of a linear-chain CRF. For time step t, Y_t is a discrete hidden state, and $\mathbf{X}_t$ is an observation data vector. The feature function set is defined over the observation vector.

For a detailed discussion on inference and training algorithms for CRFs, interested readers are referred to [27]. In the following section, we describe the task-specific details of our GPS anomaly detection component using linear-chain CRFs.

3.2 Feature Extraction

Given raw GPS data streams from taxi GPS devices, our anomaly detection component intends to automatically identify taxis with abnormal driving behaviors. The value of hidden states $\mathbf{Y}$ is thus limited to $\{abnormal, normal\}$ (More states could be included if the input data provides more information for reliable labeling). The observation sequence $\mathbf{X}$ is derived from the raw GPS streams. Specifically, at each time step, incoming raw data items are preprocessed for fast fetching, and a new observation vector $\mathbf{X}_t$ is extracted for every taxi. The observation vector is derived from data with a time stamp inside a time window and a spatial location inside a neighborhood of the target taxi. A detailed description of the feature functions used in our CRF model is given in the following part of the section. We divide a GPS data stream into non-overlapping segments, and take the observation vectors within each segment as

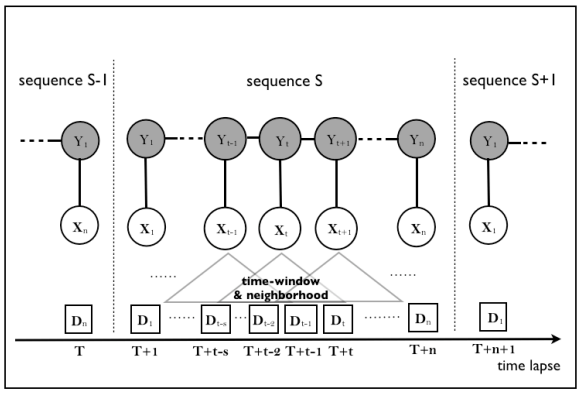


Figure 4: Illustration of data organization for CRF-based labeling. A GPS data stream is divided into non-overlapping segments, and the observation vectors within each segment are taken as an input sequence to the CRF model. D_t is the GPS raw data at time step t, and $\mathbf{X}_t$ is the derived observation vector. The output state sequence $\mathbf{Y}$ is the anomaly detection result.

an input sequence to the CRF model. The segment length is configurable, in our system, we use 200 seconds as a typical setting. For each input sequence, the computed hidden states are used as the predicted labels of the taxi over a corresponding time interval. Figure 4 illustrates how the data is organized and processed in the CRF framework.

To a large extent the performance of a CRF model is determined by the selection of the feature function set. In our GPS anomaly detection, the features at a time step are based on speed, time, location and passenger loading information.

• **speed**: Speed is a primary source of cues that indicate driving behaviors. Instead of simply taking the speed of the target taxi at a single time step alone as a feature function, we rely on statistical properties, including histograms of the target taxi's speed in a time window $[T-s,T]$, where T is the current time step, and histograms of the speed of taxis in a neighborhood of the target taxi at T. Speed is decomposed into magnitude and direction. Therefore there are four histograms in total, the histogram of speed magnitude in a time window, the histogram of speed direction in a time window, the histogram of speed magnitude in a spatial neighborhood, and the histogram of speed direction in a spatial neighborhood. Each histogram has 20 bins, and every bin of a histogram defines a feature function. Specifically, for a histogram of speed magnitude, the magnitude is discretized into 20 intervals, and each bin represents one interval. Similarly, for a histogram of speed direction, the direction is also discretized into 20 intervals, and each bin represents an angular interval of 18 degrees. The count (height) of a bin is the number of occurrences with the corresponding attribute interval. In addition to individual histograms, a mean histogram of each of the four types is also calculated over all taxis; for each taxi, the *Earth Mover's Distance* (EMD) [14] between its individual histogram and the corresponding mean histogram is taken as an additional feature function. These EMDs serve as a global measurement of how much an individual histogram deviates from the overall average.

• **time**: Time is also a very important feature that implies the likelihood of anomaly occurrence along the day. In GPS traces, time is represented in second. In the feature definition, time values are discretized into the following intervals: *morning, morning_rush_hour, noon, afternoon, afternoon_rush_hour, night, late_night*.

• **location**: Location in the raw GPS data is expressed as (longitude, latitude) coordinates. We partition the target urban area into a rectangular grid, and use the indices of the grid blocks as location values. Intuitively each grid block serves as a district in the urban area, and the feature induction and selection algorithm ([13] section 6) of the CRF model are used to extract the hidden correlations between districts and the likelihood of anomaly occurrence, even though we don't have an explicit information about the districts. Note that this feature selection algorithm works not just for location feature but for all the features.

• **load**: Boolean value indicating whether a taxi is loaded with passengers. This is also taken as a potential feature.

4 Active Learning

Active learning is a general learning paradigm and many variations exist. The active learning scenario we have in the system is most related to *pool-based active learning*, where the learner proactively chooses a sample set of data items from a (usually very large) set of unlabeled data as the candidate training set to limit the amount of labeled data required for the learner to reach an acceptable level of accuracy [23, 5] or an increased level of generalization [3, 26]. Pool based active learning is a practical and effective learning method in many applications where unlabeled data is easily obtainable while manual labeling is expensive. This is exactly the case in our system. We have an unlimited amount of streaming data while manual labeling on such data is laborious. The success of an active learning procedure depends on the sample selection criteria. In the literature, many different criteria have been proposed including label uncertainty [26] and prediction variance [6].

4.1 Sampling Criteria

In our system, we adapt the criteria proposed in [28] for CRF-related sample selection. These criteria take into account sample *uncertainty*, *representativeness*, and *diversity* to choose non-redundant samples of high uncertainty as the training set. Among the three, uncertainty measures the level of confidence a CRF model labels a data sequence. Representativeness and diversity measure the similarity among samples. The active learning procedure first chooses a set of candidate samples with the highest uncertainty, and then uses the representativeness criterion to refine the candidate set by filtering out redundant data items in the candidate set. Last, we use the diversity criterion to select from the candidate set the ones that have not yet been covered in the training set. In the following we discuss our adapted version of these sample selection criteria.

4.1.1 Uncertainty

A conditional random field provides a natural confidence measurement of the prediction it makes. For a sequence $\mathbf{x}$, the confidence (conditional probability) of label y_i of $\mathbf{x}_i$ $P(y_i|\mathbf{x})$ can be efficiently computed using the forward/backword algorithm [28]. The overall confidence $C(\mathbf{y}|\mathbf{x})$ of the label sequence $\mathbf{y}$ given input sequence $\mathbf{x}$ is defined to be the minimum confidence among all labels in the sequence, i.e., $C(\mathbf{y}|\mathbf{x}) = \min_i P(y_i|\mathbf{x})$. Given the definition of the confidence of a sample sequence $\mathbf{x}$, the uncertainty measure is defined as

$$Uncertainty(\mathbf{x}) = 1 - C(\mathbf{y}|\mathbf{x}).$$

4.1.2 Representativeness

For the subset of data sequences with high uncertainty $\mathbf{S}$, a representativeness measure is defined over each sequence $\mathbf{S}_i$ as following,

$$Representativeness(\mathbf{S}_i) = \frac{1}{|\mathbf{S}|-1}\sum_{j=1,j\neq i}^{|\mathbf{S}|} 1 - Sim(\mathbf{S}_i,\mathbf{S}_j),$$

where $Sim(\mathbf{S}_i,\mathbf{S}_j)$ is the similarity between two sequences. Given two sequences $\mathbf{S}_i =< p_{i1},...p_{im} >$ and $\mathbf{S}_j =< p_{j1},...p_{jm} >$ (p_{ik} is

the k-th data item in sequence $\mathbf{S}_i$, m is the length of a sequence. Note that in our system all sequences are of the same length), $Sim(\mathbf{S}_i, \mathbf{S}_j) = \frac{1}{m}\sum_{k=1}^{m}\cos(p_{ik}, p_{jk}) = \frac{1}{m}\sum_{k=1}^{m}\frac{p_{ik}\cdot p_{jk}}{\|p_{ik}\|\|p_{jk}\|}$, which is the average pairwise *cosine similarity* over the entire sequences. A high representativeness value means that a sample sequence is not similar to any other sequence in the candidate set. Note that the more complicated calculation of representativeness in a general context [28] has been simplified in our system because the sequences are of the same length.

We use the following formula

$$L(\mathbf{S}_i) = 0.6Uncertainty(\mathbf{S}_i) + 0.4Representativeness(\mathbf{S}_i)$$

to choose a candidate training set whose combined score $L(\mathbf{S}_i)$ exceeds a prescribed threshold, where the coefficients are empirical values determined by experiments.

4.1.3 Diversity

Once the candidate set with the highest combined scores have been chosen, we use the diversity measure to remove items that are redundant with respect to data items that are already in the training set from the previous iteration. Specifically, for each of the sequences $\mathbf{S}_i$ in the candidate set, we add it to our final training set if the similarity score between $\mathbf{S}_i$ and any item currently in the training set is not greater than $\eta = (1 + avgSim)/2$, where $avgSim$ is the average pairwise similarity among all sequences currently in the training set.

In our GPSvas system, the CRF model uses an ample set of low level features of attribute values summarized over a spatial and temporal window, while the visualization interface exhibits higher level visual cues which are easily interpretable by human intelligence, for example, abnormal speed variation patterns, irregular shapes of trajectory, strange spatial distributions of neighboring taxis, etc. The active learning module helps to improve model performance in the sense that it drives a training process that takes human labeled data, which serve as a message of human judgements made from high level visual cues, and reveals the hidden patterns from the data in the language of low level features used in the machine model. In other words, end-users (experts on urban traffic monitoring) label abnormal driving behaviors from the visualized traffic flows based on their expertise and discretion; the training procedure then induces the relevant feature set and adjust model parameters to distinguish anomalies from the normal.

5 Visualization and Interaction

Visualization and interaction play a critical role in our system. It connects the back-end machine learning components with human analysts who monitor and guide the system's execution. Three major functionalities are accommodated in the visualization and interaction components: 1) The visualization engine displays taxi trajectories and their associated text annotations generated from the anomaly detection component. This provides a general impression of the traffic flow and individual driving behaviors, for example, possible traffic jam zones and aggressive passing behaviors, to the system user; 2) In addition to the above basic visualization functionality, our system can also visualize the internal feature values that the CRF model relies on to automatically label a vehicle, i.e., the most critical features that vote for or against the decision on an anomaly detection label. This information helps analysts gain additional insights on the streamed GPS data; 3) The human-computer interaction interface allows the user to select specific information to explore and to provide guidance to the underlying machine learning models.

Note that these three functionalities are organized as integrated visualization and interaction components of the system. They coordinate with each other for presenting data and information, and conveying human knowledge and guidance to the machine models. The following three subsections discuss these three functionalities respectively.

We use the Prefuse visualization toolkit [10] for our visualization task. Due to data confidentiality issues, map is not used in the visualization interface.

5.1 Visualizing GPS traces

The GPS traces of the taxis pass through the system as data streams. These data streams are scanned only once and kept in the system for a while before being discarded. Visualization of such GPS traces includes updating the location of the taxis, displaying a partial trajectory of each taxi as well as visually presenting other information that is available in the trace records, such as whether a taxi is loaded with passengers. We also base on the statistics of speed in a time window to highlight potential anomalies.

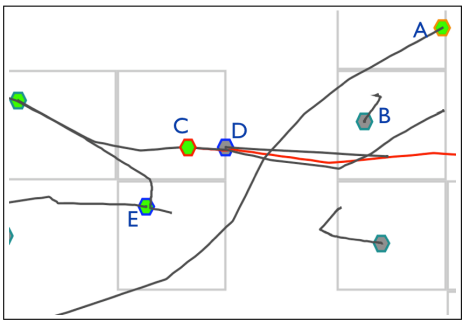


Figure 5: A visual representation of GPS traces. Taxi A is loaded with passengers, and is highlighted with an orange stroke color to signal a potential anomaly. Taxi B is not loaded with passengers. Taxi C is selected by the mouse and highlighted with a red stroke color. Nearby taxis of the selected taxi has a blue stroke color. The trajectory of the selected taxi is highlighted in red.

• **taxi trajectory**: According to the sampling rate of the GPS data, the actual position of each taxi is updated every 10-20 seconds. Directly connecting these position updates with line segments would produce unnatural zigzagging trajectories. We use a Cardinal spline [25] to generate a plausible trajectory given the position samples. Since the original GPS position samples are not sufficiently dense, to avoid unrealistic undulations in the resulting spline, we choose an appropriate tension parameter for the Cardinal spline. Afterwards, the approximate position of the taxi at any time can be computed using this spline. This generates more continuous and natural trajectories of the taxis and produces smoother vehicle movements. The actual look of the taxi trajectories interpolated by Cardinal splines can be found in Figure 5. Note that the length of the partial trajectory of a taxi is determined by a fixed-size causal time window. A longer partial trajectory indicates a higher average speed in the time window.

• **passenger loading**: Each taxi is visually represented as a thick solid dot. Its filled color is used to indicate whether a taxi is loaded with passengers or not. Specifically, a green filled dot indicates the taxi is loaded, and black indicates the opposite.

• **potential anomalies**: In this part of visualization, we use a simple cue, speed magnitude, for detecting potential abnormal driving behaviors in an unsupervised way. Specifically, we calculate a histogram of speed for every taxi within a causal time window, and compute the Earth Mover's Distance (EMD) between this histogram and the average histogram among all taxis. We also build a prior Gaussian model over the EMDs. Histograms with an EMD

to the average histogram larger than twice the standard deviation of the Gaussian (2-sigma rule) are considered potential anomalies. We use the stroke color of the solid dot to distinguish the potential anomalies from others. Orange is used for highlighting taxis with potential anomalies, and cyan is used for the others. In addition, at every time step, the taxi whose histogram has the largest EMD to the average histogram is highlighted by scaling its dot size by a factor of 3.

Such potential anomalies may be different from the anomalies labeled by the CRF model built using supervised training. These potential anomalies focus on fast moving taxis, which exhibit potentials to have true anomalous situations, while the anomalies labeled by the CRF model are more reliable because they are decided using a richer set of features with a sufficiently trained CRF model.

An example snapshot (part of the full screen) of trace visualization is shown in Figure 5.

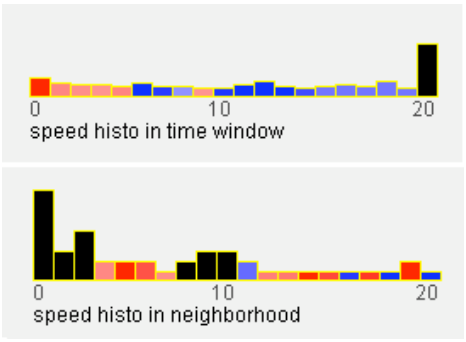


Figure 6: **Top**: the speed histogram for a time window, **Bottom**: the speed histogram for a spatial neighborhood. Each bucket in the histograms corresponds to a feature with the feature value represented by the height. Colors are used to reveal the degree of correlation (positive or negative) with the predicted label. Warm colors (red) show positive correlations while cool colors (blue) show negative correlations. neutral (black) colors shows weak correlations.

5.2 Visualizing CRF Features

In a typical application, a CRF model is used as a black box: a user trains the model weights with a set of training data, and uses the trained model to make predictions on unlabeled data afterwards. It is usually unclear to a user how the predictions are made. More specifically, what are the critical factors that contribute to the model's output. In our visualization component, we developed a module that visualizes the internal information of the CRF model to illustrate how a decision is made inside the CRF model. The internal information includes the current state of features and their weights. Visualizing such information provides the possibility that if inappropriate weights are found in the CRF model, one could tune those weights in the right direction by adding specific labeled training data through the active learning procedure. To the best of our knowledge, this is the first attempt to visualize the internal states of CRF models.

A CRF model consists of two parts: the feature set, and the weights associated with the features. In practical applications, the feature set tends to be very large such that it would be impossible to gain understanding into the model by directly displaying them in plain texts. In our visualization component, we use visual representations of the features and their weights instead. When a user selects a specific data item, the subset of features that is turned on for the specific data item are visualized, as well as the associated weights. For example, for a specific taxi at time step t, an "on" feature could be "the number of other nearby taxis whose speed is lower than 10 is between 5 and 10". In the next time step, this feature would probably become "the number of other nearby taxis whose speed is lower than 10 is less than 5". From this example we see that for different taxis or different time steps, each feature takes potentially a different value. In our visualization scheme, a feature is represented by a rectangular bar, the height of which encodes its value (count), and the color of which encodes the weight associated with the feature. Positive weights are shown as red while negative weights are shown as blue. A linear interpolation is used to obtain colors for intermediate weights. Figure 6 shows an example of the visualization of a feature set consisting of bins in the speed histograms. Representative abnormal cases and their histograms are shown in Figure 7.

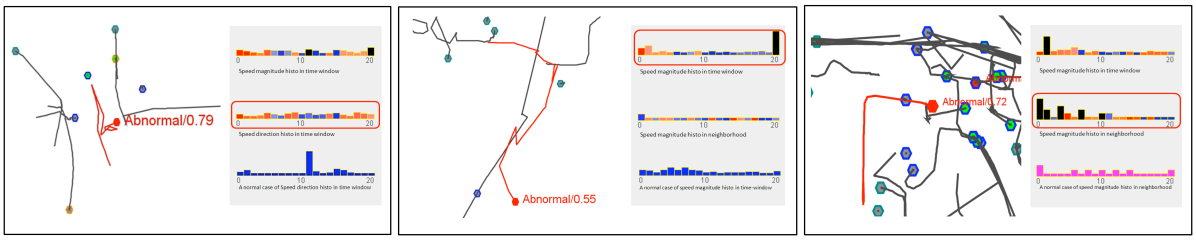
5.3 Interaction Interface

Human-machine interaction is another indispensable part of our system. In our system, interaction is bi-directional: on one hand users explore the visualized taxi partial trajectories and their associated text labels indicating whether any anomalies have been automatically detected; on the other hand, the underlying active learning module proactively selects items with highly uncertain labels and requests feedbacks from human experts. Human experts give responses by manually providing labels to the requested items. Such labels are used for training an improved version of the CRF model. To provide appropriate user control, the interface allows the users to customize most of the system parameters, such as radius of neighborhood, size of objects, type of information to display, etc.

To accommodate different types of user interaction described above, our system is designed to have three interaction modes: *basic mode*, *monitoring mode* and *tagging mode*. The basic mode only visualizes the raw GPS traces without any labels, and users can only perform basic exploration. In the monitoring mode, the anomaly detection component is activated, and anomaly tags are shown dynamically on the screen. Users can also choose to view the internal CRF states of the tagged data items. In the tagging mode, the active learning module is activated. Highly uncertain labels from the CRF model are highlighted, requesting for user input. CRF model training with the newly labeled data is also activated in the tagging mode. In the following we describe each of these three modes and their corresponding interactions.

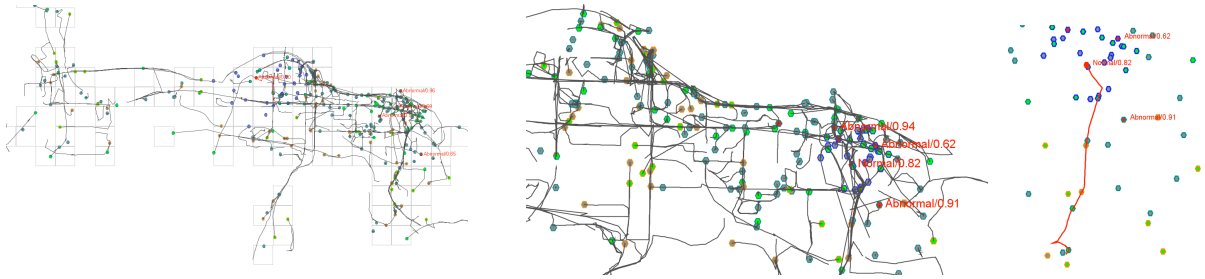
• **basic mode**: Basic interactions in this mode includes: (a) *zoom-in/zoom-out*, which is controlled by mouse scroll, (b) *dragging*, which translates the center of the view port on the 2D plane by direct mouse dragging, (c) *taxi selection*, which highlights the selected taxi whose detailed information is also displayed on screen, (d) *replay*, which goes back in time to show the data that has just passed by. This allows users to check important scenarios of the traffic when necessary. There are a few other interaction operations such as pause/resume, change neighborhood radius for extracting the neighborhood features, information filtering to control the set of visual items (trajectories, grid) to be rendered on screen, and change the size of the fonts and taxi items, etc. Note that all the operations in this mode are available in the other modes as well. Figure 8 shows the zoom-in affects with various visualization settings at different levels.

• **monitoring mode**: The extra computation in the monitoring mode is running the anomaly detection component. Feature sequences of the taxis are periodically fed to the CRF model, which returns automatically tagged results to the visualization engine, which adds text annotations to the taxis if being labeled as anomalies. The anomalies are highlighted by setting the filled color to red. The annotation for a detected anomaly consists of a text string and an associated confidence value. In this mode, internal CRF feature values and weights that contribute to a labeling result can be visualized when the analyst selects the specific taxi which the labeling result is associated with.



(a) (b) (c)

Figure 7: Representative abnormal cases and histograms. (a) An abnormal case with an irregular pattern of driving directions. (b) An abnormal case involving high speed. (c) An abnormal case with a crowded neighborhood (possible traffic jam). An identified critical factor is highlighted with a red rectangle with rounded corners.



(a) (b) (c)

Figure 8: Information exploration at different scales. (a) A global view of taxi trajectories with a superposed grid, (b) a zoom-in to a local area where anomalies occur, (c) further zoom-in to a finer scale to view the local taxi distribution (trajectories of unselected taxis are hidden).

• **tagging mode**: Different from the monitoring mode, the tagging mode is not for monitoring the anomalous cases but for collecting manually labeled results used in active learning. In the tagging mode, the active learning module selects representative taxis whose predicted labels are highly uncertain and send them to the visualization engine. The visualization engine shows the predicted labels and their uncertainty values together with the taxis. Human experts can select any of these items and assign it a manual label to either confirm or correct the predicted label. The number of mouse clicks is minimized in such a way that clicks are only required when the label needs to be flipped. At the end of manual tagging, the user can trigger the system to train an improved version of the CRF model using the manually labeled data. Once the training concludes, our system switches to the updated CRF model.

6 Anomaly Detection Performance

We have tested the anomaly detection performance of our GPS visual analytics system on an Intel i7-860 2.8GHz Quad Core processor. In this following we discuss the experimental setup and results on anomaly detection using CRFs.

6.1 Query by Committee

We use the query-by-committee [7] strategy instead of a single model to improve the robustness of anomaly detection. Specifically, five separate CRF models are initially trained using different sets of training sequences. Given a new sequence, each of the five models makes an independent prediction and the one with the highest confidence level is chosen as the final result. In other words, we choose the prediction result by the committee member who has the highest confidence in its decision.

6.2 Accuracy

Table 1 summarizes the prediction accuracy of the individual models, averaged result and the query-by-committee model. Prediction accuracy is shown for three different versions of each model, the initial version trained with relatively little training data, and two versions trained after the first and second round of active learning. It is obvious that active learning steadily improves labeling accuracy over iterations. Given the prediction accuracy of the five individual CRF models, their average (expected) accuracy and the final prediction accuracy of the query-by-committee model, we confirm that the query-by-committee strategy can in general improve prediction accuracy (compare the last two columns of Table 1), and can potentially achieve performance better than any of the individual models.

Table 1: Summary of labeling accuracy.

train	1	2	3	4	5	AVG	QBC
baseline	0.62	0.72	**0.77**	0.61	0.66	0.67	0.66
train1	0.83	0.73	0.87	0.79	0.77	0.80	**0.88**
train2	0.83	0.78	0.82	0.82	0.82	0.81	**0.90**

7 Related Work

GPS signal indicates the temporally changing location of the GPS device wearer. Many techniques and systems have been developed to visualize GPS or trajectory data in a 2D or 3D space. [1] and [21] are two recent works on GPS data analysis and trajectory visualization. A system is introduced in [20] for visualizing real-time multi-user GPS data from the Internet in a 3D VRML model. *GPSVisualizer*, *Google Earth*, *Yabadu Maps*, *GPS-Track-Analyse.NET* and *FUGAWI Global Navigator* are examples of online systems that support the visualization of GPS data in various applications. The prevalence of GPS devices and wearable computing devices make *wearable computing* [2] a new emerging field of research. A recent technique for visualizing aircraft trajectories has also been presented in [11].

Our system integrates GPS data visualization with anomaly detection using conditional random field models. This type of appli-

cation has not yet been found in the visual analytics literature. In machine learning, however, there is an increasing popularity in information retrieval [15], behavior classification or human activity understanding [9, 17] based on mobile sensor data, i.e., GPS data. In [15], the authors use hierarchically constructed conditional random fields to model human activities and extract significant locations in a map from GPS traces. Although [15] uses a similar type of data and learning model as we do, the difference lies in that their goal is to discover important patterns and locations in human activities while our goal is to perform anomaly detection in taxi driving behaviors. They try to develop a completely automatic method while our system is semi-automatic and human-computer interaction is essential. We adopt a visual analytics approach to integrate human expertise and achieve a good performance.

8 Conclusions and Future Work

In this paper, we have introduced a visual analytics system for anomaly detection in urban taxi GPS data streams, and demonstrated that such an approach integrates the power of machine learning models with human intelligence. Human understanding to the data and machine performance on anomaly labeling are mutually enhanced through the visualization & interaction interface and active learning procedure.

There exist a few directions for future work. First we could exploit the geographical information in the system, i.e., design features based on street information, or use a map to improve trajectory reconstruction. Second, collect user assessment to improve the system usability and make a user-friendly interface design. Third, increase the number of hidden states, such as driving skill level and regional traffic status, of the CRF model, given that GPS traces provide relevant information as a type of sensor data. Another potential direction for future work is to extend the linear-chain CRF model to more complex models. For example, a hierarchical hidden Markov model [4] allows hidden states to be defined at different levels of granularity to model a hierarchical structure, or a semi-CRF model [22], which models hidden state transitions as a semi-Markov chain. Both models have a higher computational complexity, but with careful model design, we expect better performance in label prediction.

Acknowledgements

This work was partially supported by National Science Foundation (IIS 09-14631), National Natural Science Foundation of China (60728204, 60902104), National High-tech R&D Program of China (2009AA01Z302), and Shenzhen Science and Technology Foundation (GJ200807210013A). Part of the data used in the article was provided by Shenzhen Department of Transportation.

References

[1] G. Andrienko, N. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. In *ACM SIGKDD Explorations Newsletter*, pages 38–46, 2007.

[2] M. Billinghurst, S. Weghorst, and T. Furness. Wearable computing for three dimensional CDCW. In *Proceedings of the International Symposium on Wearable Computing*, 1997.

[3] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. In *Machine Laerning*, volume 15, pages 201–221, 1992.

[4] S. Fine and Y. Singer. The hierarchical hidden markov model: Analysis and applications. In *MACHINE LEARNING*, pages 41–62. Kluwer Academic Publishers, Boston, 1998.

[5] A. Finn and N. Kushmerick. Active learning selection strategies for information extraction. In *ECML-03 Workshop on Adaptive Text Extraction and Mining*, 2003.

[6] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168, 1997.

[7] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168, 1997.

[8] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. In *unpublished manuscript*, 1971.

[9] B. L. Harrison, S. Consolvo, and T. Choudhury. Using multi-modal sensing for human activity modeling in the real world. In *Handbook of Ambient Intelligence and Smart Environments*, 2009.

[10] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proceedings of Computer Human Interaction*, pages 421–430, 2005.

[11] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. In *IEEE Transactions on Visualization and Computer Graphics*, pages 1017–1024, 2009.

[12] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *Advances in Neural Information Processing Systems 16*. MIT Press, Combridge, MA, 2003.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-2001)*, 2001.

[14] E. Levina and P. Bickel. The earthmovers distance is the mallows distance: Some insights from statistics. In *Proceedings of International Conference on Computer Vision*, pages 251–256, 2001.

[15] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26, 2007.

[16] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. In *Mathematical Programming 45*, pages 503–528, 1989.

[17] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: Scalable sound sensing for people-centric applications on mobile phones. In *The International Conference on Mobile Systems, Applications, and Services*, 2009.

[18] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of The Seventh Conference on Natural Language Learning (CoNLL-2003)*, 2003.

[19] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of The IEEE*, volume 77, 1989.

[20] I. Rakkolainen, S. Pulkkinen, and A. Heinonen. Visualizing real-time gps data with internet's VRML worlds. In *Proceedings of the 6th ACM international symposium on Advances in geographic information systems*, pages 52–56, 1998.

[21] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. Visually driven analysis of movement data by progressive clustering. In *Information Visualization*, pages 225–239, 2008.

[22] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192, 2004.

[23] M. Sassano. An empirical study of active learning with support vector machines for Japanese word segmentation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[24] K. Sato and Y. Sakakibara. RNA secondary structure alignment with conditional random fields. In *Bioinformatics*, pages 237–242, 2005.

[25] I. J. Schoenberg. *Cardinal Spline Interpolation*. Society for Industrial Mathematics, 1987.

[26] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[27] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In *Introduction to statistical relational learning, chapter 4*. MIT Press, Cambridge, MA, 2007.

[28] C. T. Symons, N. F. Samatova, R. Krishnamurthy, B. H. Park, T. Umar, D. Buttler, T. Critchlow, and D. Hysom. Multi-criterion active learning in conditional random fields. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, 2006.